Cyan Technology

eCOG1k Evaluation Kit

User Manual

V1.2

23 January 2008

Confidential and Proprietary Information

© **Cyan Technology Ltd., 2006-2008**

This document contains confidential and proprietary information of Cyan Technology Ltd. and is protected by copyright laws. Its receipt or possession does not convey any rights to reproduce, manufacture, use or sell anything based on information contained within this document.

Cyan Technology™, the Cyan Technology logo and Max-eICE™ are trademarks of Cyan Holdings Ltd. CyanIDE® and eCOG® are registered trademarks of Cyan Holdings Ltd. Cyan Technology Ltd. recognises other brand and product names as trademarks or registered trademarks of their respective holders.

Any product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by Cyan Technology Ltd. in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. Cyan Technology Ltd. shall not be liable for any loss or damage arising from the use of any information in this guide, any error or omission in such information, or any incorrect use of the product.

This product is not designed or intended to be used for on-line control of aircraft, aircraft navigation or communications systems or in air traffic control applications or in the design, construction, operation or maintenance of any nuclear facility, or for any medical use related to either life support equipment or any other life-critical application. Cyan Technology Ltd. specifically disclaims any express or implied warranty of fitness for any or all of such uses. Ask your sales representative for details.

Revision History

Version	Date	Notes
V1.0	24/01/2006	First release.
V1.1	30/03/2007	Updated for CyanIDE V1.4.
V1.2	23/01/2008	Changed front page.

Contents

List of Tables iv
1 Introduction 1
1.1 Evaluation Kit Contents 1
1.2 Requirements 1
1.3 Additional Documents 1
1.4 Part Identification 1
2 Quick Start 2
2.1 Set Up System 2
2.2 Copy the Example Projects 2
2.3 Running an Example Project 2
3 Software 3
3.1 CyanIDE 3
3.2 USB eICE Drivers 3
4 Installing CyanIDE 4
4.1 From the Cyan Tools CD-ROM 4
4.2 From the Cyan Website 4
4.3 Installing the USB eICE Driver 4
5 CyanIDE Examples 5
5.1 Evaluation Board Examples 5
5.2 Simulator Examples 10
6 Evaluation Board 11
6.1 Overview 11
6.2 Description 11
7 Connections and Jumper Links 12
7.1 J5: CPU Power 12
7.2 J10-J13: CPU Pins 12
7.3 J2: Expansion Connector 13
7.4 J14, J15: Serial Ports 13
7.5 J23-J26: Analogue Input Potentiometers 13
Appendix A Important Notes 14
Appendix B Evaluation Board Layout 15

List of Tables

1: J10-J13: CPU pins. 12
2: J2: Expansion Connector 13
3: J14, J15: Serial ports. 13

1 Introduction

1.1 Evaluation Kit Contents

- eCOG1k Evaluation Board (eCOG1 LCEval Board Issue B)
- USB cable (A to B)
- Null modem serial cable
- CD-ROM containing CyanIDE development software and documentation

1.2 Requirements

- A Windows-based PC system.
  (minimum 1GHz CPU speed, higher speed recommended).
- Windows 2000 or Windows XP operating system.
- 100MB free disk space.
- 512MB memory (1GB recommended).
- A spare USB port.
- A spare serial port (optional).
- System administrator privileges are required for software installation.

1.3 Additional Documents

1. eCOG1k User Manual
2. CyanIDE User Manual
3. eCOG1k Evaluation Board Issue B Circuit Diagram
4. TN006 Configuring the FTDI FT2232

1.4 Part Identification

In this document, any reference to eCOG1 means the generic chip and is applicable to all versions. All eCOG1 devices are suffixed according to their version; any reference to a particular version such as eCOG1k is specific to that version.

2 Quick Start

Please also read *Appendix A Important Notes*.

2.1 Set Up System

- Unpack and check contents of kit.
- Install CyanIDE development software and USB device drivers.
  See section 4 *Installing CyanIDE* for more details.
- Connect the USB cable between the evaluation board and the host PC.

2.2 Copy the Example Projects

CyanIDE includes a range of example projects for the development and evaluation kits, copied during installation into the <*examples*> directory under the CyanIDE install directory, usually <*C:\Program FIles\Cyan Technology\CyanIDE*>. The installation process also creates a <*CyanIDE Projects*> directory in the user's *My Documents* folder. It is recommended that the examples for the eCOG1k evaluation kit are copied into this projects directory before use, to avoid making any changes to the original examples.

- Open the *My Documents* directory and browse into the *CyanIDE Projects* directory. This includes a shortcut to the CyanIDE examples directory.
- Open the <*CyanIDE Examples*> shortcut and then the examples directory for this evaluation board <*eCOG1k eval board\general*>.
- Type **ctrl-A** or click **Edit->Select All** to select all the eCOG1k example projects, then type **ctrl-C** or click **Edit->Copy** to copy them to the clipboard.
- Click the **Back** button three times to return to the CyanIDE Projects directory.
- Type **ctrl-V** or click **Edit->Paste** to paste the example projects from the clipboard into the current directory.

2.3 Running an Example Project

As an introduction, try the LED example project located in the <*led*> directory.

- Start CyanIDE.
- From the main menu, select **Project->Open** and browse to the LED example directory <*CyanIDE Projects\led*>. Select the project file <**.cyp*> and click **Open**. CyanIDE loads the project and displays the files included in the project in the navigator pane at the left of the main window.
- Select **Build->Rebuild All** from the main menu. This compiles the project source files and links the object code into a download image file.
- Select **Debug->Run**. CyanIDE connects to the eCOG1k target processor on the evaluation board, downloads the application code and begins execution.

3 Software

3.1 CyanIDE

The CyanIDE software development package supports the eCOG1 family of microcontrollers, providing project management, source code editor, C compiler, assembler, linker, source level debugger, and online help files. It is available on CD-ROM, or as a download to registered users on the Cyan web site at www.cyantechnology.com.

CyanIDE communicates with the eCOG1k microcontroller on the target system via the eICE debug port. The evaluation kit incorporates the circuit for the Cyan USB eICE adaptor, a separate debug adaptor is not required.

CyanIDE includes a range of example applications for the evaluation board. These can provide a good starting point for customer applications, or just as examples showing how to set up a software project. Further application examples including CyanIDE project files with source code are available on the web site support pages.

3.2 USB eICE Drivers

The software drivers for the USB eICE adaptor are included in the CyanIDE development package. CyanIDE V1.4 or later includes the required version of the driver files as standard.

To ensure that the driver files are present, install the CyanIDE software and any necessary updates before connecting the evaluation board to the host PC. Further details about the software installation are shown later in this document.

Note that any previous version of CyanIDE should be uninstalled before the latest version is installed. This includes any USB device drivers for Cyan products such as the evaluation board, which should be removed via the Device Manager. Installing the new version of CyanIDE also reinstalls the USB eICE device driver.

4 Installing CyanIDE

4.1 From the Cyan Tools CD-ROM

Insert the CD into the CD-ROM drive. The html start page should load automatically in the default browser.

If the start page does not load automatically, or Autorun is disabled for the CD-ROM drive, then Browse the Cyan CD in a file manager or explorer window, and open the file <*index.htm*>.

When the start page is displayed, click on the **Menu** button, then select the **Install CyanIDE** item to begin installation of the development software. Follow the instructions presented by the installation program.

4.2 From the Cyan Website

- Navigate to the software downloads page on the website, located at http://www.cyantechnology.com/support/updates.php.
- Download the CyanIDE full version installation file to a temporary directory. Please note that users must log in to the website with their registered account name and password for the support forum to download this file.
- Execute the downloaded file to install the development software. Follow the instructions presented by the installation program.

4.3 Installing the USB eICE Driver

CyanIDE V1.4 includes the driver for the USB eICE adaptor and evaluation board as standard, and it is not necessary to install it separately.

When an eICE adaptor or evaluation board is connected to a PC for the first time, the USB enumeration process identifies it as new hardware, and Windows starts the *Found New Hardware* process. Driver installation under Windows 2000 and Windows XP is fully automatic and does not require any user interaction.

5 CyanIDE Examples

CyanIDE includes a number of example software projects, pre-configured for use with the eCOG1k evaluation board, the eCOG1k development board or the eCOG1 simulator. The examples for the eCOG1k evaluation board are described briefly here.

5.1 Evaluation Board Examples

The following example applications for this version of the evaluation board are included as standard with CyanIDE V1.4.

Example	Comments
adc_test	Using the analogue-to-digital converter
cache	Demonstrates various cache modes
clock	Implements a time-of-day clock using low-power sleep mode
duart_irq	DUART serial port using interrupts and circular buffers
eval_pcb	The software shipped on the evaluation board
ForthCLI	A simple FORTH Command Line Interpreter
irom1k	A memory-resident application to program the internal flash memory
led	Uses a timer to flash LEDs on the development board
perfect	Searches for perfect numbers

Many examples use one of the serial ports to report results or display messages. Use a terminal program such as 'HyperTerminal' to communicate with the application. The default serial port configuration is shown below.

- 9600 Baud
- 8 data bits
- no parity bits
- one stop bit
- no flow control

5.1.1 adc_test

The 'adc_test' application demonstrates reading analogue input values via the ADC. The example uses counter CNT1 as a timer to trigger readings at a preset interval and averages the results over a number of readings. It uses DUART channel A to implement an interrupt-driven serial port for both standard input and output. Entering one of the characters 'A', 'R', 'S', 'T' or 'V' changes the selection and configuration of the ADC input channels. The values are output via the serial port.

In order to use the stdio.h functions such as `printf()` and `scanf()`, users must supply versions of `putchar()` and `_getchar()`. These functions are contained in the main application file.

File	Comments
adc.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
adc.c	Application code
devboard.cfg	Peripheral configuration file
internal.map	Memory map

5.1.2 cache

The 'cache' application runs a quick benchmark program with different cache configurations and displays achieved performance values based on the execution time for each configuration. The performance figures are reported via DUART channel A. The TMR timer is used to measure the execution times of the benchmark routines. This code demonstrates the following features:

- Use of the cache
- Interrupts in C
- Clock configuration in the SSM (System Support Module)
- Use of the TMR timer
- Use of the DUART serial port

File	Comments
cache.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
cache.c	Application code
devboard.cfg	Peripheral configuration file
internal.map	Memory map

5.1.3 clock

The 'clock' application outputs a time value (hh:mm:ss) every second. The time is displayed on the LCD and echoed via DUART channel A. Counter CNT1 is used as a timer to generate an interrupt at 1 second intervals. On each interrupt, the processor is woken from the sleep state and set to stay awake. The main routine outputs the time and puts the CPU back to sleep. This code demonstrates the following features.

- Interrupts in C
- Sleep mode
- Deactivating clocks for sleep mode
- Waking up using peripheral interrupts
- Clock configuration in the SSM
- Use of counter CNT1 as a timer
- Use of the DUART serial port

For further information on low power operation refer to Application Notes AN004 and AN008 on the Cyan CD-ROM.

File	Comments
clock.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
clock.c	Application code
putchar.c	Character output routine
devboard.cfg	Peripheral configuration file
internal.map	Memory map

5.1.4 duart_irq

The 'duart_irq' application demonstrates how to use the DUART for an interrupt-driven serial port with circular buffers. Standard input and output is through DUART channel A. All characters that the application receives are echoed back. In order to use the stdio.h functions such as `printf()` and `scanf()`, users must supply versions of `putchar()` and `_getchar()`. These functions are contained in the main application file.

File	Comments
duart.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
duart.c	Application code
devboard.cfg	Peripheral configuration file
internal.map	Memory map

5.1.5 eval_pcb

The 'eval_pcb' application is shipped with the evaluation board. The software demonstrates the following:

- The DUART serial port
- GPIO for inputs and outputs
- Reading analogue inputs via the ADC
- Using counter CNT1 as a timer
- Interrupts in C

One of two tests is selected by setting DIP switch 1, and results are output via DUART channel A.

File	Comments
eval_pcb.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
adc.c/h	ADC code
isr.c	Interrupt handlers
main.c	C entry point code
util.c/h	Utility routines
welcome.c/h	Welcome message code
evalboard.cfg	Peripheral configuration file
internal.map	Memory map

5.1.6 ForthCLI

The 'ForthCLI' application implements a simple command line interpreter, using the Forth language. It is described in more detail in Application Note AN020.

File	Comments
ForthCLI.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
main.c	Application code
devBoard.c/h	Board-specific functions
Kernel.c/h	Forth CLI kernel
Memory.c/h	Variables and definitions for the Forth memory, stacks, dictionary, etc.
RS232.c/h	Interrupt-driven serial port using DUART A.
StdDefs.h	Common definitions
ecog1.cfg	Peripheral configuration file
internal.map	Memory map

5.1.7 irom1k

The 'irom1k' application demonstrates how to program the internal flash memory of the eCOG1k. The program runs in internal RAM and so the MMU must be set after the program has been downloaded to map the IRAM internal ram area as CODE space. This is performed using a Python initialisation function that is installed when then project is loaded.

A program similar to this one is used by CyanIDE to program the eCOG1k when a `<project>.rom` file is downloaded.

For further information on programming the internal flash memory, refer to Application Notes AN001, AN018 and AN019 on the Cyan CD-ROM. For information and examples on programming external flash memory, refer to Application Note AN002.

File	Comments
irom1k.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
flash.c	Main application code
fmc.c/h	Flash programming code
tim.c/h	Timing functions
initfunc.py	Python script code to set up the MMU. The **Run on project load** option for this file is set.
iram.map	Memory map

5.1.8 led

The 'led' application uses counter CNT1 as a timer to control the flashing of four LEDs connected to GPIO output signals.

File	Comments
led.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
led.c	Application code
devboard.cfg	Peripheral configuration file
internal.map	Memory map

5.1.9 perfect

The 'perfect' application searches for perfect numbers and outputs them via DUART channel A. A perfect number is one whose divisors sum to itself.

File	Comments
perfect.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
perfect.c	Application code
putchar.c	Character output routine
devboard.cfg	Peripheral configuration file
internal.map	Memory map

5.2 Simulator Examples

The following example simulator applications are also included with CyanIDE. They can be found in the *<examples\simulator>* directory located below the CyanIDE installation directory.

Example	Comments
primes	Searches for prime numbers
oscillator	Example of using a simulated memory-mapped device

5.2.1 primes

The 'primes' application searches for prime numbers and displays results in the Debug Output Window. The results are displayed using an implementation of `putchar()` that includes the assembler `PRINT` instruction. The simulator interprets this instruction according to the **Print mode** option in **Project Properties**: printing a character, a hexadecimal value, or raising an exception.

File	Comments
primes.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
primes.c	Application code
putchar.c	Character output routine
simulator.map	Memory map

5.2.2 oscillator

The 'oscillator' application reads a simulated ADC, which is generating data representing a sine wave, and writes a filtered value back to a memory location. The filtered value is logged to a file.

File	Comments
oscillator.cyp	CyanIDE project file
cstartup.asm	C environment initialisation
irq.asm	Entry point and interrupt vectors
oscillator.c	Application code
putchar.c	Character output routine
oscillator.py	Simulated ADC written in Python. The **Run on project load** option for this file is set.
simulator.map	Memory map - includes CUSTOM entry

6 Evaluation Board

6.1 Overview

The evaluation board has the following major features.

- eCOG1k microcontroller with 32Kwords internal flash memory.
- USB interface to eICE debug port.
- Two RS-232 serial ports.
- Four 12-bit analogue inputs.
- Eight user/status LEDs.
- Eight user switches.
- Piezo sounder.
- Various pin headers for input and output ports.
- Powered from the USB port.

6.2 Description

The eCOG1k evaluation board provides a very simple target system for experimentation with hardware and software for the eCOG1k microcontroller. It has a USB interface to the eICE debug port on the eCOG1k, suitable for direct connection to the host PC. It is supported by the CyanIDE development package which includes a number of example software projects configured for the evaluation board.

The eCOG1k microcontroller has a 16-bit CPU architecture and a wide range of on-chip peripherals. It operates at clock speeds of up to 25MHz internally from a 5MHz crystal or 32.768kHz watch crystal. Refer to the eCOG1k User Manual for further details.

Two serial ports are available on the evaluation board. Both are fitted with a 9-way D-type plug and are wired as DTE ports. A null-modem serial cable is provided to connect a PC serial port to either port.

The four 12-bit analogue inputs on the eCOG1k device can be connected individually to onboard potentiometers VR1-VR4 for testing, or to 0V, by fitting jumper links to pin headers J23-J26.

The evaluation board is powered from the USB eICE debug connection.

The eight LEDs are connected to the eCOG1k on port pins A_0 to A_7. DIP switch SW2 is connected on port pins C_0 to C_3, and SW3 is connected on port pins J_2 to J_5. Leave the switches in the off position to allow the use of these port pins elsewhere if required.

7 Connections and Jumper Links

7.1 J5: CPU Power

The 3.3V power supply for the eCOG1k microcontroller is connected through header J5. A link must be fitted to J5 for normal operation.

This allows the supply current drawn by the eCOG1k to be measured. Remove the link, and connect an ammeter between J5 pins 1 and 2. The ammeter measures the supply current to the eCOG1k. Note that in standby or low power operating modes, the supply current is comparable with the input and output leakage currents. This makes it difficult to measure the supply current accurately in low power modes when the eCOG1k is connected to other active devices.

7.2 J10-J13: CPU Pins

All the CPU pins are accessible on pin headers J10-J13.

J10			
PortE 7	1	2	PortE 6
PortE 5	3	4	PortE 4
PortE 3	5	6	PortE 2
PortE 1	7	8	PortE 0
GND	9	10	CPU break
VCC3.3	11	12	GND
PortK 7	13	14	PortK 6
NC	15	16	VCC3.3
GND	17	18	NC
PortK 5	19	20	PortK 4
VCC3.3	21	22	GND
eICE MISO	23	24	eICE MOSI
eICE Clock	25	26	eICE LOADB
eICE CS	27	28	GND
GND	29	30	GND
GND	31	32	AGND

J11			
VCC3.3	1	2	Vin1
Vin2	3	4	Vin3
Vin4	5	6	Reset Out
Reset In	7	8	PortA 0
PortA 1	9	10	PortA 2
PortA 3	11	12	PortA 4
PortA 5	13	14	PortA 6
PortA 7	15	16	VCC3.3
GND	17	18	PortB 0
PortB 1	19	20	PortB 2
PortB 3	21	22	PortB 4
PortB 5	23	24	PortB 6
PortB 7	25	26	PortC 0
PortC 1	27	28	PortC 2
PortC 3	29	30	PortD 0
GND	31	32	PortD 1

J12			
PortD 2	1	2	PortD 3
PortL 0	3	4	PortL 1
PortL 2	5	6	PortL 3
PortL 4	7	8	PortL 5
PortL 6	9	10	PortL 7
Vcc3.3	11	12	GND
PortK 0	13	14	PortK 1
NC	15	16	GND
VCC3.3	17	18	NC
PortK 2	19	20	PortK 3
PortJ 0	21	22	PortJ 1
PortJ 2	23	24	PortJ 3
PortJ 4	25	26	PortJ 5
PortJ 6	27	28	PortJ 7
PortI 7	29	30	PortI 6
PortI 5	31	32	PortI 4

J13			
PortI 3	1	2	PortI 2
PortI 1	3	4	PortI 0
VCC3.3	5	6	GND
PortH 7	7	8	PortH 6
PortH 5	9	10	PortH 4
PortH 3	11	12	PortH 2
PortH 1	13	14	PortH 0
PortG 7	15	16	PortG 6
PortG 5	17	18	PortG 4
PortG 3	19	20	PortG 2
PortG 1	21	22	PortG 0
VCC3.3	23	24	GND
PortF 7	25	26	PortF 6
PortF 5	27	28	PortF 4
PortF 3	29	30	PortF 2
PortF 1	31	32	PortF 0

Table 1: J10-J13: CPU pins

7.3 J2: Expansion Connector

Pin header J2 provides access to the unused signals on the FTDI USB device.

J2			
BDBUS0	1	2	ACBUS0
BDBUS1	3	4	ACBUS1
BDBUS2	5	6	ACBUS2
BSBUS3	7	8	ACBUS3
BDBUS4	9	10	VCC3.3
BDBUS5	11	12	GND
BDBUS6	13	14	GND
BDBUS7	15	16	GND
NC	17	18	NC
BCBUS0	19	20	GND
BCBUS1	21	22	GND
BCBUS2	23	24	GND
BCBUS3	25	26	GND
SI/WUB	27	28	GND
NC	29	30	NC
PWREN#	31	32	GND

Table 2: J2: Expansion Connector

7.4 J14, J15: Serial Ports

J14 and J15 are standard PC compatible 9-way D-type serial connectors. Both are fitted with a plug, wired as a DTE, and require a null-modem cable to connect to a PC. J15 also provides support for RTS/CTS hardware handshake signals using two GPIO lines.

J14				J15			
GND	5			GND	5		
		9	NC			9	NC
NC	4			NC	4		
		8	NC			8	CTS
TX	3			TX	3		
		7	NC			7	RTS
RX	2			RX	2		
		6	NC			6	NC
NC	1			NC	1		

Table 3: J14, J15: Serial ports

7.5 J23-J26: Analogue Input Potentiometers

The eCOG1k analogue input signals may be connected to onboard potentiometers or to GND for testing or experiments. Each of the four analogue inputs has its own potentiometer, and a pin header that selects where it is connected. For analogue input VIN1, link J23 pins 1-2 to connect it to the potentiometer VR1, or link pins 2-3 to connect it to GND. If no links are fitted then no onboard signal is connected to VIN1, and the analogue input may be connected to external signals via header J31.

Similarly, analogue inputs VIN2-4 are configured with headers J24-26.

Appendix A Important Notes

The following recommendations should be observed when using the USB eICE interface.

- Connect the USB cable from the eICE interface directly to the host PC, not via an external hub. CyanIDE can fail to restart the eICE debug connection after any errors if the USB device is connected via an external hub.
- The memory window in CyanIDE can be quite slow to refresh across the USB eICE link. Close the memory window when it is not required to improve the speed of response to commands.
- Do not disconnect the USB eICE cable or power down the target system while CyanIDE is running. This can cause CyanIDE to hang up on the next attempt to connect to the target system via eICE.
  Ensure that CyanIDE is closed down before disconnecting the USB cable or powering down the target system.
- CyanIDE may report an error message on its first attempt to connect to the target system via eICE. This occurs when it tries to find a connection to a target system on the parallel port instead of the USB port. Repeat the command and CyanIDE should connect to the USB eICE target system successfully.
  If the debugger still does not start, check that power is present on the target system, that all required jumper links are fitted, and that the USB cable is connected correctly to the host PC.

Appendix B Evaluation Board Layout